

A Physarum-inspired model for the probit-based stochastic user equilibrium problem

Shuai Xu^a, Wen Jiang^{a,*}

^a*School of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shannxi, 710072, China*

Abstract

Stochastic user equilibrium is an important issue in the traffic assignment problems, tradition models for the stochastic user equilibrium problem are designed as mathematical programming problems. In this article, a *Physarum*-inspired model for the probit-based stochastic user equilibrium problem is proposed. There are two main contributions of our work. On the one hand, the origin *Physarum* model is modified to find the shortest path in traffic direction networks with the properties of two-way traffic characteristic. On the other hand, the modified *Physarum*-inspired model could get the equilibrium flows when traveller's perceived transportation cost complies with normal distribution. The proposed method is constituted with a two-step procedure. First, the modified *Physarum* model is applied to get the auxiliary flows. Second, the auxiliary flows are averaged to obtain the equilibrium flows. Numerical examples are conducted to illustrate the performance of the proposed method, which is compared with the Method of Successive Average method.

Keywords:

Traffic assignment problem, user equilibrium, elastic demand, Physarum, network

*Corresponding author: School of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shannxi, 710072, China. Tel:+86 029 88431267; fax:+86 029 88431267. E-mail address: jiangwen@nwpu.edu.cn

1. Introduction

The traffic assignment problem (TAP) refers to assign traffic trip of each origin-destination (OD) pair to the links in the transportation networks and give an OD trip matrix [1, 2, 3, 4, 5, 6]. Traditionally, the TAP problem falls into two major classes, known as user equilibrium(UE) and stochastic user equilibrium(SUE) [7, 8]. Considering the negative effect of road traffic congestion upon travel time, the user equilibrium was conceptualised by Wardrop [9]. Assuming that the travellers know the precise route cost and choose the route with minimum cost, the UE principle is reached when no traveller can reduce transportation cost by changing routes. To overcome the unrealistic assumption of precise perception of route travel time across travellers, the stochastic user equilibrium was firstly defined by Daganzo and Sheffi [10]. The SUE principle is obtained when no traveller's perceived transportation cost can be reduced by unilaterally changing routes.

In the exist literatures, the SUE problem was classed into two types: the logit-based SUE and the probit-based SUE, according to those random costs following Gumble or normal distribution [11]. Due to explicit form and calculation, the logit-based SUE model has paid great attention [12, 13, 14, 15, 16, 17] . However, the probit-based SUE model behaves more appealing, attributing to the fact that it can take no account of overlapping, or correlated routes [18]. Sheffi and Powell proposed the well-known Method of Successive Average (MSA) with predetermined step sizes for solving the probit-based SUE problem[19]. Maher modified the model to decrease the computation complexity by choosing the optimal step length along the search direction[18]. Though distributed computing approaches were executed to reduce the computing time for the probit-based SUE problem [20], the development of probit-based SUE models still has some limitations, especially on large networks, which can be explained by the difficulty of completing path enumeration or Monte Carlo simulation.

Considering that performing the SUE principle in conventional probit-based models is difficult, we propose solving the stochastic traffic assignment problem by a *Physarum*-inspired model. The plasmodium of *Physarum* polycephalum is a large amoeboid organism, which contains a great number of nuclei and tubular structures [21]. These tubular structures will distribute

protoplasm as a transportation network. Recently, it is shown that *Physarum* has the capacity of finding the short path between two points in a given labyrinth [22]. Tero *et al.* [23] inspired an mathematical model that can capture the basic dynamics of network adaptability through iteration of local rules and produces solutions with properties comparable to or better than those of real-world infrastructure networks. Bonifaci [24] has proved that the mathematical model can convergence to the shortest path. Later, the *Physarum* model was used to design and simulate transport network [25, 26, 27, 28], find the short path [29, 30, 31, 32]. To handle the uncertainty in the real application [33], the *Physarum* model can also solve shortest path under uncertain environment[34, 35].

Considering the continuity of the flow and protoplasmic network adaptivity, now the UE problem can also be solved by the *Physarum* model [36]. Note that UE problem is just a subsection of the SUE problem, here we present a *Physarum*-inspired model for the probit-based SUE problem. In the proposed model, the origin *Physarum* model is modified to adapt the directed network with multiple sources and directions and the link travel time is regard as the length of *Physarum* tubular structures.

This paper is organized as follows. In Section 2, the SUE assignment problem in traffic networks is reviewed and the *Physarum* polycephalum model is briefly introduced. In Section 3, a *Physarum*-inspired model for the SUE problem is presented. In Section 4, numerical examples are given to prove the rationality and convergence properties of the proposed model. Finally, the paper ends with conclusions in Section 5.

2. PRELIMINARIES

In this section, the basic theories, including the probit-based stochastic user equilibrium problem and *Physarum* polycephalum model, are briefly introduced.

2.1. Probit-based user problem

2.1.1. Notations, assumptions and definitions

Given a strongly connected transportation network $G = (N, A)$, where N and A denote the sets of nodes and links, respectively. Network attributes are denoted by notations as follow:

R	Set of origin nodes, $R \subseteq N$
S	Set of destination nodes, $S \subseteq N$
r	An origin node, $r \in R$
s	An destination node, $s \in S$
K_{rs}	Set of all the paths between OD pair rs.
q_{rs}	Travel demand between OD pair rs, and all the OD travel demands are grouped into column vector, $\mathbf{q} = (\dots, q_{rs}, \dots)^T$, $r \in R, s \in S$
f_k^{rs}	Traffic flow on path k between OD pair rs, $k \in K_{rs}$.
\mathbf{f}^{rs}	Column vector of traffic flows on the paths between OD pair rs, $\mathbf{f}^{rs} = (\dots, f_k^{rs}, \dots)^T$, $k \in K_{rs}$.
\mathbf{f}	Column vector of traffic flows on the all paths, $\mathbf{f} = (\dots, \mathbf{f}^{rs}, \dots)^T$, $r \in R, s \in S$.
c_k^{rs}	Travel time on path k between OD pair rs, $k \in K_{rs}$
\mathbf{c}^{rs}	Column vector of traffic time on the paths between OD pair rs, $\mathbf{c}^{rs} = (\dots, c_k^{rs}, \dots)^T$, $k \in K_{rs}$
\mathbf{c}	Column vector of traffic time on the all paths, $\mathbf{c} = (\dots, \mathbf{c}^{rs}, \dots)^T$, $r \in R, s \in S$
x_a	Traffic flow on link a, $a \in A$
\mathbf{x}	Column vector of all link flows, $\mathbf{x} = (\dots, x_a, \dots)^T$, $a \in A$.
t_a	Asymmetric travel time on link a, $a \in A$.
\mathbf{t}	Column vector of all the link-travel-time functions, $\mathbf{t} = (\dots, t_a, \dots)^T$, $a \in A$

$$\begin{aligned}
\delta_{a,k}^{rs} & \quad \delta_{a,k}^{rs} = 1 \text{ if } k \in K_{rs} \text{ between OD pair } rs \text{ traverses link } a \in A, \\
& \quad \delta_{a,k}^{rs} = 0, \text{ otherwise.} \\
\Delta^{rs} & \quad \text{link/path incidence matrix associated with OD pair } rs, \Delta^{rs} = \\
& \quad (\delta_{a,k}^{rs}, a \in A, k \in K_{rs}) \\
\Delta & \quad \text{link/path incidence matrix for the entire network, } \Delta = \\
& \quad (\dots, \Delta^{rs}, \dots)
\end{aligned}$$

According to the cost flow superposition principle, the path travel time can be valued as the summation of link travel time [19], which can be expressed as:

$$c_k^{rs} = t_a \cdot \delta_{a,k}^{rs} \quad (1)$$

Compactly, the relation between path travel time and link travel time can be expressed in vector form, namely:

$$\mathbf{c} = \Delta^T \cdot \mathbf{t} \quad (2)$$

Assuming that the network users perceived link travel time is consist of the determined link travel time and random error term. The perceived link travel time is thus expressed as:

$$T_a = t_a(x_a) + \varepsilon_a, \quad \forall a \quad (3)$$

where link travel time function $t_a(x_a)$ is positive, continuously differentiable and strictly monotone increasing. The error term ε_a associated with link a is a normally distributed random variable with zero mean for the probit-based SUE problem [19], which can be expressed as following:

$$\varepsilon_a \sim N(0, \gamma t_a^0), \quad \forall a \quad (4)$$

where γ is a proportionality constant parameter and t_a^0 is a constant which usually equals free-flow link. Similarly, the perceived link travel time is also a normally distributed random variable, namely:

$$T_a \sim N(t_a, \gamma t_a^0), \quad \forall a \quad (5)$$

Due to the linearity of the incidence relationships, the perceived path travel time also follows a multivariate normal distribution leading to the probit model for the path choice, which can be expressed as:

$$C_k^{rs} = \sum_a T_a \cdot \delta_{a,k}^{rs}, \quad \forall r, s, k \in K_{rs} \quad (6)$$

Obviously, according to the accumulation of random variables, C_k^{rs} is also a normally distributed random variable with c_k^{rs} mean :

$$c_k^{rs} = \sum_a t_a \cdot \delta_{a,k}^{rs}, \quad \forall r, s, k \in K_{rs} \quad (7)$$

Let f_k^{rs} denote the traffic flow on path k between OD pair rs , it can be expressed as the following equation:

$$f_k^{rs} = q_{rs} \cdot P_k^{rs}, \quad \forall r, s, k \in K_{rs} \quad (8)$$

where P_k^{rs} denotes the path choice probability for path k between OD pair rs . On the basis of economics principles, P_k^{rs} denotes the probability of path k being the shortest one for given path travel time between OD pair rs [37], namely:

$$P_k^{rs} = P(C_k^{rs} \leq C_l^{rs}, \forall l \neq k), \quad \forall r, s, k \in K_{rs} \quad (9)$$

The stochastic user equilibrium is reached when no user can reduce his perceived travel time by unilaterally changing routes [10]. The objective function of SUE problem was first proposed and proved by Sheffi and Power [19]:

$$\min Z(x) = - \sum_{rs} q_{rs} \cdot E[\min_{k \in K_{rs}} \{C_k^{rs}\} | \mathbf{c}^{rs}(\mathbf{x})] + \sum_a \{x_a \cdot t_a(x_a) - \int_0^{x_a} t_a(w) dw\} \quad (10)$$

where we have used the results of Williams that

$$\frac{\partial}{\partial c_k^{rs}} E[\min_{k \in K_{rs}} \{C_k^{rs}\} | \mathbf{c}^{rs}(\mathbf{x})] = P_k^{rs} \quad (11)$$

2.1.2. The Method of Successive Average

The MSA algorithm developed by Sheffi and Powell [19] was the first algorithm applied to solve the SUE problem. In the MSA process, the link costs are calculated by the current link flows. An auxiliary link flow pattern is produced through a stochastic network loading procedure. And the search direction is obtained by the difference between the auxiliary link flow and the current link flow. The step size is predetermined by a descent sequence with respect to the iterations. The procedures of MSA method are summarized as following:

Step 1.1: Choose initial link travel costs $\{t_a^0, \forall a\}$, usually free-flow costs. Find an initial feasible flow pattern $\{x_a^1, \forall a\}$ by carrying out, for example, a pure stochastic loading using mean costs. Set the iteration count n to 1.

Step 1.2: According to the current flow pattern $\{x_a^n, \forall a\}$, calculate the current travel costs $\{t_a^n(x_a^n), \forall a\}$.

Step 1.3: Given the mean travel costs $\{t_a^n, \forall a\}$ and the demands of OD pairs, find the auxiliary flow pattern $\{\hat{y}_a^n, \forall a\}$ by carrying out a pure stochastic loading.

Step 1.4: Calculate the new current solution according to the equation:

$$x_a^{(n+1)} = x_a^n + \frac{1}{n}(\hat{y}_a^n - x_a^n), \quad \forall a \quad (12)$$

Step 1.5: Convergence test. If the following condition is fulfilled, then stop and output . Otherwise, $n = n + 1$, go to step 2.

$$\sqrt{\sum_{a \in A} (x_a^{(n+1)} - \hat{y}_a^n)^2} \leq \varepsilon_0 \quad (13)$$

The search direction is found by using the auxiliary flow pattern $\{\hat{y}_a^n, \forall a\}$, which is computed through Monte Carlo simulation methods:

- Step 2.1: Initialize counter $i = 1$.
- Step 2.2: Sample one realization from each link, using $f(T_a|t_a^n)$.
- Step 2.3: Assign "all or nothing" from each origin to each destination. This results in the auxiliary flow pattern $y_a^{(i)}$.
- Step 2.4: Average the flow for each link, $\bar{y}_a^{(i)} = [(i-1)\bar{y}_a^{(i-1)} + y_a^{(i)}]/i$.
- Step 2.5: If the stopping criterion is met, set $\hat{y}_a^n = \bar{y}_a^{(i)} \forall a$; If not, set $i = i + 1$ and go to step 2.2.

where $f(T_a|t_a^n)$ is the probability density of T_a , and the travel time of link a can be calculated according to Eq.(5) The stopping criterion referred to at step 2.5 may be based on the reduction of the variance of $\bar{y}_a^{(i)}$ as i grows [7, 19], such as a fixed number of drawings I_0 .

2.2. *Physarum polycephalum* model

Physarum polycephalum is a single-celled amoeboid organism, which is also called as plasmodium in the vegetative phase. It is able to solve the shortest path selection, basing on its special foraging mechanism: the transformations of tubular structures and a positive feedback from flow rates. The high rates of the flow motivate tubes to thicken, and the diameter of the tube minishes at a low flow rate. A total introduction for the *physarum polycephalum* model is given below.

Supposing the shape of the network formed by the *Physarum* represented by a graph, plasmodial tube refers to an edge of the graph and a junction between tubes refers to a node. Assuming a set of nodes N , N_1 and N_2 are signed as the source and destination nodes, any others are labeled as N_3, N_4, N_6, N_7 , etc. The edge connecting nodes N_i and N_j is remarked as M_{ij} . The flux from node N_i to node N_j through edge M_{ij} is remarked as Q_{ij} , which we can expressed as [23] :

$$Q_{ij} = \frac{\pi r_{ij}^4}{8\eta L_{ij}}(p_i - p_j) = \frac{D_{ij}}{L_{ij}}(p_i - p_j) \quad (14)$$

where η is the viscosity of the fluid and $D_{ij}=\pi r_{ij}^4/8\eta$ is measure of the conductivity of the edge M_{ij} tube. p_i is the measure of the pressure at the node N_{ij} and L_{ij} is the length of the edge of M_{ij} . According to the conservation law of flow, the inflow and outflow must be balanced, namely:

$$\sum Q_{ij}=0, \quad (j \neq 1, 2) \quad (15)$$

For the source nodes N_1 and N_2 , the flux equations can be denoted as:

$$\sum_i Q_{i1} + I_0 = 0 \quad (16)$$

$$\sum_i Q_{i2} - I_0 = 0 \quad (17)$$

where I_0 is the flux from the source node to the destination node, which is assumed as a constant in the model. According to the Eqs (14)-(17), the network Poisson equation for the pressure is derived as following:

$$\sum_i \frac{D_{ij}}{L_{ij}}(p_i - p_j) = \begin{cases} -1 & \text{for } j = 1, \\ +1 & \text{for } j = 2, \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

by further setting $p_2 = 0$ as the basic pressure level, the pressure of all nodes can be determined according to Eq.(18) and all Q_{ij} can also be determined by solving Eq.(14).

To accommodate the adaptive behavior of the plasmodium, the conductivity D_{ij} is assumed to change when adapting to the flux Q_{ij} . And tubes with zero conductivity will die out. The conductivity of each tube is described as the following equation [23]:

$$\frac{d}{dt}D_{ij} = f(|Q_{ij}|) - \alpha D_{ij} \quad (19)$$

where α is the decay rate of the tube and f is monotonically increasing continuous function which satisfies $f(0) = 0$. The *Physarum* can converge to the shortest path when $f(|Q_{ij}|) = |Q|$ and $\alpha = 1$ [24]. Obviously, the positive feedback exists in the model.

3. PROPOSED METHOD

In this section, we employ the proposed *Physarum* model to solve the stochastic user equilibrium problem. Generally speaking, there are three problems to be addressed:

1. The original *Physarum* model is used to solve the shortest path problem in undirected graphs ($L_{ij} = L_{ji}$) while most network is directed graphs ($L_{ij} \neq L_{ji}$) in real traffic assignment problem.
2. There is only one source node in the shortest path finding mode, but we should solve the traffic assignment problem with multiple sources and sinks.
3. The modified *Physarum* model should approach the optimal flow distribution in the traffic assignment problem.

3.1. *Physarum*-based model for the shortest path in the directed network

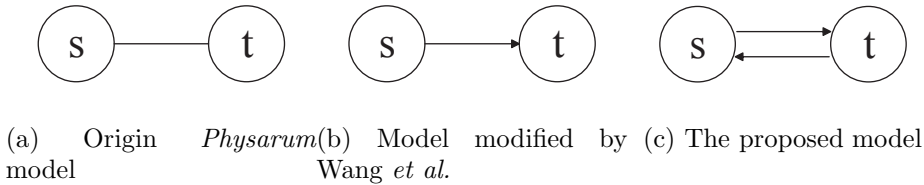


Figure 1: Three *Physarum* models

In the original *Physarum* model, each arc shown in Figure 1.a is bidirectional, which means the distance from node i to node j is same as that from node j to node i . To solve the constrained shortest path problem, Wang *et al.* [30] proposed a modified *Physarum* model shown in Figure 1.b where each edge is regarded as two tubes with opposite direction and equal weight. And there is only one direction between two nodes, which means that the flux can flow from node s to node t . The modified *Physarum* model has the ability to find the shortest path in the directed network. While most roads in the city

have the properties of two-way traffic characteristic and opposite directions are separated with each other, the *Physarum* model modified by Wang *et al.* couldn't work in these networks. Here we proposed a new modified *Physarum* model shown in Figure 1.c. There are two opposite directions between node s and node t , and the length of two opposite directions is denoted by L_{st} and L_{ts} . Basing on the feature of foraging behavior, the conductivity matrix D implies not only the conductivity but also the direction of each tube, namely $D_{ij} \neq D_{ji}$ during iterations. In order to implement this idea into the original *Physarum* model, Eq. (18) is modified as following:

$$\sum_i \left(\frac{D_{ij}}{L_{ij}} + \frac{D_{ji}}{L_{ji}} \right) (p_i - p_j) = \begin{cases} -1 & \text{for } j = 1, \\ +1 & \text{for } j = 2, \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

To keep the validity of conductivity, the conductivity equation defined in Eq.(14) should be improved as following:

$$Q_{ij} = \begin{cases} \frac{D_{ij}}{L_{ij}}(p_i - p_j), & \frac{D_{ij}}{L_{ij}}(p_i - p_j) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (21)$$

Particularly when $L_{st} = \inf$ or $L_{ts} = \inf$, which means that the flux can only flow from node t to node s or from node s to node d , our modified model is the same as that of Wang *et al.* Exactly, the model modified by Wang *et al.* is a section of our modified model.

3.2. *Physarum*-based model for multiple sources and directions

In the original *Physarum* model, there is only one source node and one direction node. While in the stochastic user equilibrium problem, there are always multiple OD pairs. Assuming O denoting the set of origin nodes, $O \subseteq N$, and D denoting the set of destination nodes, $D \subseteq N$, we can modify Eqs. (16) and (17) as following:

$$\sum_i Q_{io} + I_o = 0, \quad o \in O \quad (22)$$

$$\sum_i Q_{id} - I_d = 0, \quad d \in D \quad (23)$$

where I_o is the inflow at the origin node o , I_d is the outflow at the destination node d . To ensure the flow is distributed in a optimal way, here we use the modified model proposed by Zhang [28] to replace Eq.(18):

$$\sum_i \left(\frac{D_{ij}}{L_{ij}} + \frac{D_{ji}}{L_{ji}} \right) (p_i - p_j) = \begin{cases} -I_o, & \forall o \in O, \\ +I_d, & \forall d \in D, \\ 0, & otherwise \end{cases} \quad (24)$$

3.3. *Physarum-inspired model for the probit-based SUE problem*

Now, we study how to solve the probit-based stochastic user equilibrium problem basing the *Physarum*-inspired model. Due to the feature of foraging behavior, the flow and the conductivity along each link are continuous in the process of *Phyasrun* approaching the shortest path. While in other classical shortest path algorithms, such as Dijkstra algorithm [38], Floyd algorithm [39], algorithms approach the shortest path by traversing all the nodes until the destination node is visited, which is totally uncontinuous.

Considering the continuity and dynamic reconfiguration of *Physarum* model, we can update the link travel time within each iteration. The flux will be redistributed by the modified *Physarum* model when the link travel time is updated during iterations. Here we adopt the following equation to update the length of link a :

$$C_a^n = \frac{C_a^{n-1} + t_a(x_a)}{2} \quad (25)$$

where x_a denotes the traffic flow on link a at the $(n)_{th}$ iteration, C_a^n and C_a^{n-1} represent the length of link a at the n_{th} and $(n-1)_{th}$ iteration. And the search direction of link length C_a is guided by $t_a(x_a)$. Note that in equilibrium, there will be $C_a = t_a(x_a)$, which means the length of link a equals the travel time along link a .

The main steps of the proposed method for the probit-based stochastic user equilibrium problem is presented in Algorithm 1. In the process of Monte Carlo simulation, we can use *Physarum* model to replace "All or nothing" method to calculate auxiliary flow pattern. In the iteration, C^n is the current link travel cost matrix at the n iteration, C_{ij}^n represents the current travel

cost from node i to node j , at the n iteration. Particularly, C^0 is the free-flow link travel cost matrix, C_{ij}^0 represents the free-flow travel cost from node i to node j .

Different from the MSA algorithm, the current travel cost(C_{ij}^n) is calculated by the modified auxiliary flow in the proposed algorithm, attributed to the continuity of *Physarum* model. During the the process of Monte Carlo simulation the flow and the conductivity along each link are continuous.

More importantly, the current travel cost C_{ij}^n doesn't equal C_{ji}^n when $\hat{Q}_{ij}^{(I_0)} \neq \hat{Q}_{ji}^{(I_0)}$, which means the same edge have different travel costs in two opposite directions. This peculiarity is rather important in nowadays traffic network. Because most roads in the city don't interfere in two opposite directions, opposite directions are separated with each other. Indeed, the flow in edge L_{ij} does't influence the travel cost in edge L_{ji} .

4. NUMERICAL EXAMPLES

In this section, two examples are designed to prove the rationality and convergence properties of the proposed algorithm, a one source and sink node network and a multiple sources and sinks network. The inner iteration I_0 and the outer iteration n are compared with those in the MSA algorithm.

Both tests are investigated using a simple network shown in Figure 2, which is introduced by Sheffi and Powell [19]. Link costs are calculated by the US Bureau of Public Roads (BPR) function, which is expressed as following:

$$t_a(x_a) = \alpha_a + \beta_a x_a^4 \quad (26)$$

where parameters of the link cost functions for each link, α_a and β_a are shown in Table 1.

According to Eq.(5), the value of γ determines the variance of the perceived link travel cost ,which has a great effect on the convergence properties of the algorithm. In both tests, γ is kept constant at 0.3. In Eq.(13), ε_0 is the condition of stopping iteration, if the value of ε_0 is to small, the process of

Algorithm 1 a Physarum-inspired model for the probit-based SUE problem

```

//  $\varepsilon_0$  is the stopping criterion of the whole method.
//  $I_0$  is the stopping criterion of the Monte Carlo simulation, also called as
the inner iteration.
//  $n$  is called as the outer iteration.
//  $\bar{Q}^n$  is the current flow matrix at the  $n$  iteration,  $\bar{Q}_{ij}^n$  represents the
current flow from node  $i$  to node  $j$ .
//  $\hat{Q}^{(I_0)}$  is the modified auxiliary flow matrix.
 $D_{ij} = [0.5, 1](\forall i, j = 1, 2, \dots, N \wedge C_{ij}^0 \neq 0)$ 
if  $C_{ij}^0 == inf$  then
     $D_{ij} = 0$ 
end if
 $Q_{ij} = 0(\forall i, j = 1, 2, \dots, N)$ 
 $\bar{Q}_{ij}^0 = 0(\forall i, j = 1, 2, \dots, N)$ 
 $p_{ij} = 0(\forall i, j = 1, 2, \dots, N)$ 
 $n = 1$  //Iteration counter
while  $\varepsilon \leq \varepsilon_0$  do
     $C_{ij}^n = \frac{C_{ij}^{n-1} + t_{ij}(\hat{Q}_{ij}^{(I_0)})}{2}(\forall i, j = 1, 2, \dots, N)$ 
     $i = 1$  //Monte Carlo simulation counter
    while  $i \leq I_0$  do
         $L_{ij} = N(C_{ij}^n, \gamma C_{ij}^0) \quad (\forall a \in A)$  //Using Eq.(5)
        //Calculate the pressure of every node using Eq.(24)

$$\sum_i \left( \frac{D_{ij}}{L_{ij}} + \frac{D_{ji}}{L_{ji}} \right) (p_i - p_j) = \begin{cases} -I_o, & \forall o \in O, \\ +I_d, & \forall d \in D, \\ 0, & otherwise \end{cases}$$

        //Calculate the flux of every edge using Eq.(21)

$$Q_{ij} = \begin{cases} \frac{D_{ij}}{L_{ij}}(p_i - p_j), & \frac{D_{ij}}{L_{ij}}(p_i - p_j) > 0 \\ 0, & otherwise \end{cases}$$

 $D_{ij}^{i+1} = (D_{ij}^i + Q_{ij})/2$ 
 $\hat{Q}^{(i)} = [(i-1)\hat{Q}^{(i-1)} + Q^{(i)}]/i$ 
         $i = i + 1$ 
    end while
     $\bar{Q}^n = [(n-1)\bar{Q}^{n-1} + \hat{Q}^{(I_0)}]/n$ 

$$\varepsilon = \sqrt{\sum_{i,j \in N} (\bar{Q}_{ij}^n - \bar{Q}_{ij}^{n-1})^2}$$

     $n = n + 1$ 
end while

```

Table 1: Parameters of the link cost functions for each link, $t_a(x_a) = \alpha_a + \beta_a x_a^4$

From node	TO node	α_a	β_a
1	2	20	0.0056
1	5	18	0.0078
2	1	20	0.0071
2	6	19	0.0033
2	3	23	0.0086
3	2	23	0.0108
3	7	16	0.0101
3	4	17	0.0063
4	3	17	0.0116
4	8	22	0.0138
5	1	18	0.0131
5	6	14	0.0093
5	9	24	0.0026
6	2	19	0.0048
6	5	14	0.0041
6	7	17	0.0123
6	10	20	0.0056
7	3	16	0.0078
7	6	17	0.0071
7	8	13	0.0033
7	11	26	0.0086
8	4	22	0.0108
8	7	13	0.0101
8	12	19	0.0063
9	5	24	0.0016
9	10	7	0.0138
10	9	7	0.0131
10	6	20	0.0093
10	11	18	0.0026
11	10	18	0.0048
11	7	26	0.0141
11	12	17	0.0123
12	8	19	0.0056
12	11	17	0.0078

both algorithms cost much calculating time. However, we can't get the final travel flux if its value is too large. Hence, in order to compare the speed of convergence, the value of ε_0 is kept constant at 0.1 in the proposed algorithm and the MSA algorithm. And all computational experiments are executed using Matlab on Intel(R) Core(TM) i5-5200U processor (2.2Ghz) with 8.00

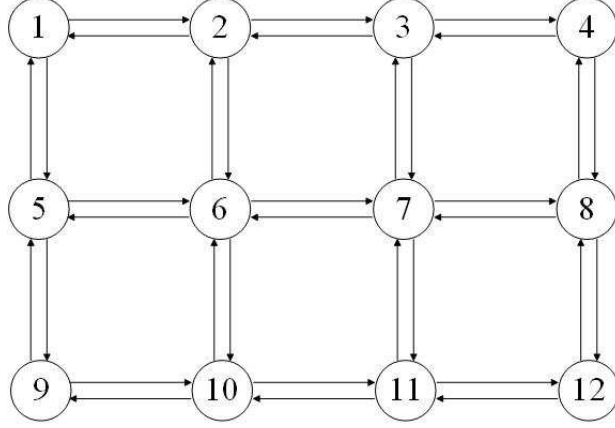


Figure 2: A simple network with 12 nodes used in [19]

GB of RAM under Win 8.

4.1. Example 1

In this example, there is only one source-direction from node 1 to node 12 with travel rate of $q_{1,12} = 20$ vehicles per unit time. To study the effect of the inner iteration I_0 , we examined the flow on particular link corresponding the number of inner iterations. Here, we choose the traffic flow of link $L_{6,7}$ in both algorithms. The effect of inner iterations is illustrated in Figure 3.

It's clearly that the convergence per equilibrium iteration improves when the inner iteration I_0 augments. However, in the proposed algorithm, the deviation between lower iteration and equilibrium flow is obviously smaller than that in the MSA algorithm when the counter of outer iterations is small.

What's importantly, both algorithms can get the same equilibrium flow when the inner iterations are different. Sheffi and Powell has proved that very few inner simulation iterations (possibly just one) may be sufficient to achieve a reasonable convergence rate of the equilibrium iterations by the SMA algorithm [19]. Note that the proposed algorithm can also get the similar result when the inner iteration equals 1 in Figure 3, we speculate that one iteration can also achieve a reasonable convergence rate of the equilibrium iterations. The numerical example is presented as below.

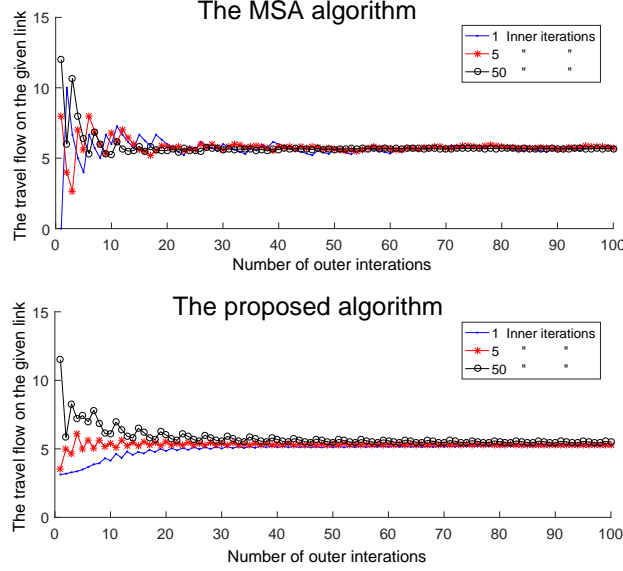


Figure 3: Convergence of the proposed algorithm vs the MSA algorithm

To prove the rationality and convergence properties of the proposed algorithm, the inner iteration L_0 should be kept same in both algorithms. So we should keep the inner iteration I_0 equal 1 in both algorithm. The results of link traffic flow calculated by both algorithms are shown in Table 2.

It's obviously there are same traffic paths in the network. And traffic rates in each link are almost similar calculated by both algorithms, which differ by no more than 0.32 vehicles per unit time. Besides, the convergence rate of the proposed algorithm is faster than that of the SMA algorithm. This peculiarity will become much more important especially when the network is quite large. Considering that responsiveness of the traffic flow assignment is much more significant in nowadays traffic network, the proposed algorithm contributes a positive idea to reduce computing time.

Table 2: The link traffic flow calculated by both algorithms in 4.1

From node	TO node	The MSA algorithm	The proposed algorithm
outer iteration n		12233	236
computer time(s)		2.317002	0.069317
1	2	10.3639	10.2070
1	5	9.6361	9.5445
2	1	0	0
2	6	4.4459	4.4894
2	3	5.9180	5.7079
3	2	0	0
3	7	2.7803	2.5665
3	4	3.1377	3.1324
4	3	0	0
4	8	3.1377	3.1328
5	1	0	0
5	6	4.9213	4.7524
5	9	4.7148	4.7896
6	2	0	0
6	5	0	0
6	7	5.6918	5.4874
6	10	3.6754	3.7607
7	3	0	0
7	6	0	0
7	8	7.6230	7.5404
7	11	0.8492	0.5210
8	4	0	0
8	7	0	0
8	12	10.7607	10.6752
9	5	0	0
9	10	4.7148	4.7948
10	9	0	0
10	6	0	0
10	11	8.3902	8.5612
11	10	0	0
11	7	0	0
11	12	9.2393	9.0669
12	8	0	0
12	11	0	0

4.2. Example 2

In this example, we also used the traffic network shown in Figure 2 and Table 1. Different from Example 1, the origin-destination demands are assumed as $q_{1,12} = 10$ and $q_{1,8} = 10$, which denote the rate of vehicles per unit time. The inner iteration was set as 10 in this example. The results of link traffic flow calculated by both algorithms are shown in Table 3. Clearly, traffic rates calculated by both algorithms are also similar and the proposed method obviously get the equilibrium flow faster than the MSA algorithm. The maximum error of the link flows is no more than 0.35 vehicles per unit time. The computing time don't increase compared with that in Example 1.

5. CONCLUSIONS

Considering of deviation between traveller's perceived transportation cost and actual cost, the stochastic user equilibrium is much more significant than user equilibrium. This paper presents a *Physarum*-inspired model for the probit-based stochastic user equilibrium problem. The *Physarum* model is modified to solve the SUE problem in the first time. To satisfy the characteristic of the real traffic networks, the origin *Physarum* model is modified to find the shortest path in direction networks with multiple sources and directions. Considering the foraging behavior of *Physarum*, the *Physarum* could find the shortest travel time path between each OD pair. The equilibrium flows could be obtained when *Physarum* couldn't find a shorter travel time path.

We compared the proposed algorithm with the MSA algorithm. And Numerical results showed that the proposed algorithm can effectively achieve the SUE solution in practice. If the inner iteration properly assigned, the proposed algorithm is faster and more efficient than the MSA algorithm. Note that many investigations about paralleled *Physarum* model have been achieved [40], the time consumption of the proposed algorithm will obviously reduced in concurrent computation. Besides, the proposed method is easy to combine with other algorithms [41].

Table 3: The link traffic flow calculated by both algorithms in 4.2

From node	TO node	The MSA algorithm	The proposed algorithm
outer iteration n		686	179
computer time(s)		2.162422	0.068314
1	2	10.3988	10.1945
1	5	9.6058	9.4830
2	1	0	0
2	6	3.6292	3.5431
2	3	6.7686	6.6450
3	2	0	0
3	7	2.2849	2.1953
3	4	4.4803	4.4454
4	3	0	0
4	8	4.4803	4.4424
5	1	0	0
5	6	5.1153	4.7598
5	9	4.4905	4.7273
6	2	0	0
6	5	0	0
6	7	6.4263	6.3017
6	10	2.3182	1.9937
7	3	0	0
7	6	0	0
7	8	8.7109	8.4797
7	11	0	0.0325
8	4	0	0
8	7	0	0
8	12	3.2044	3.0647
9	5	0	0
9	10	4.4905	4.7273
10	9	0	0
10	6	0	0
10	11	6.8088	6.7691
11	10	0	0
11	7	0	0
11	12	6.8088	6.7691
12	8	0	0
12	11	0	0

ACKNOWLEDGMENTS

The work is partially supported by National Natural Science Foundation of China (Grant No. 61671384), Natural Science Basic Research Plan in

Shaanxi Province of China (Program No. 2016JM6018), the Fund of SAST (Program No. SAST2016083), the Seed Foundation of Innovation and Creation for Graduate Students in Northwestern Polytechnical University (Program No. Z2016122).

References

- [1] D. P. Bertsekas, E. M. Gafni, Projection methods for variational inequalities with application to the traffic assignment problem, in: *Non-differential and Variational Techniques in Optimization*, Springer, 1982, pp. 139–159.
- [2] M. Papageorgiou, Dynamic modeling, assignment, and route guidance in traffic networks, *Transportation Research Part B: Methodological* 24 (1990) 471–495.
- [3] H. Yang, S. Yagar, Traffic assignment and signal control in saturated road networks, *Transportation Research Part A: Policy and Practice* 29 (1995) 125–139.
- [4] A. Ziliaskopoulos, A linear programming model for the single destination System Optimum Dynamic Traffic Assignment problem, *Transportation Science* 34 (2000) 37–49.
- [5] Y. Liu, J. Bunker, L. Ferreira, Transit Users’ Route-Choice Modelling in Transit Assignment: A Review, *Transport Reviews* 30 (2010) 753–769.
- [6] W.-B. Du, X.-L. Zhou, O. Lordan, Z. Wang, C. Zhao, Y.-B. Zhu, Analysis of the chinese airline network as multi-layer networks, *Transportation Research Part E: Logistics and Transportation Review* 89 (2016) 108–116.
- [7] Y. Sheffi, W. Powell, A comparison of stochastic and deterministic traffic assignment over congested networks, *Transportation Research Part B: Methodological* 15 (1981) 53 – 64.
- [8] M. Smith, M. L. Hazelton, H. K. Lo, G. E. Cantarella, D. P. Watling, The long term behaviour of day-to-day traffic assignment models, *Transportmetrica A-Transport Science* 10 (2014) 647–660.

- [9] J. G. Wardrop, Some theoretical aspects of road traffic research, in: Inst Civil Engineers Proc London /UK/, pp. 72–73.
- [10] C. F. Daganzo, Y. Sheffi, On stochastic models of traffic assignment, *Transportation Science* 11 (1977) 253–274.
- [11] D. Watling, User equilibrium traffic network assignment with stochastic travel times and late arrival penalty, *European Journal of Operational Research* 175 (2006) 1539–1556.
- [12] R. B. Dial, A probabilistic multipath traffic assignment model which obviates path enumeration, *Transportation Research* 5 (1971) 83–111.
- [13] C. Fisk, Some developments in equilibrium traffic assignment, *Transportation Research Part B Methodological* 14 (1980) 243–255.
- [14] M. Chen, A. S. Alfa, Algorithms for solving fisk’s stochastic traffic assignment model, *Transportation Research Part B Methodological* 25 (1991) 405–412.
- [15] T. Larsson, Z. Liu, M. Patriksson, A dual scheme for traffic assignment problems, *Optimization* 42 (1997) 323–358.
- [16] Z. Liu, S. Wang, Q. Meng, Toll pricing framework under logit-based stochastic user equilibrium constraints, *Journal of Advanced Transportation* 48 (2014) 1121C1137.
- [17] B. Zhou, M. C. J. Bliemer, M. G. H. Bell, J. He, Two new methods for solving the path-based stochastic user equilibrium problem, *Computer-Aided Civil and Infrastructure Engineering* 31 (2015) 100C116.
- [18] M. J. Maher, P. C. Hughes, A probit-based stochastic user equilibrium assignment model, *Transportation Research Part B Methodological* 31 (1997) 341–355.
- [19] Y. Sheffi, W. B. Powell, An algorithm for the equilibrium assignment problem with random link times, *Networks* 12 (1982) 191–207.
- [20] Z. Liu, Q. Meng, Distributed computing approaches for large-scale probit-based stochastic user equilibrium problems, *Journal of Advanced Transportation* 47 (2013) 553–571.

- [21] S. L. Stephenson, H. Stempen, Myxomycetes : a handbook of slime molds, *Bioscience* 45 (1995) 601–602.
- [22] T. Nakagaki, H. Yamada, Á. Tóth, Intelligence: Maze-solving by an amoeboid organism, *Nature* 407 (2000) 470–470.
- [23] A. Tero, S. Takagi, T. Saigusa, K. Ito, D. P. Bebbber, M. D. Fricker, K. Yumiki, R. Kobayashi, T. Nakagaki, Rules for biologically inspired adaptive network design., *Science* 327 (2010) 439–442.
- [24] V. Bonifaci, K. Mehlhorn, G. Varma, Physarum can compute shortest paths, in: *Acm-Siam Symposium on Discrete Algorithms*, p. 121C133.
- [25] A. Adamatzky, Bioevaluation of world transport networks, *Bioevaluation of World Transport Networks* 43 (2012) 368.
- [26] A. Adamatzky, M. Lees, P. Slood, Bio-development of motorway network in the netherlands: a slime mould approach, *Advances in Complex Systems* 16 (2013) 1250034.
- [27] V. Evangelidis, M. A. Tsompanas, G. C. Sirakoulis, A. Adamatzky, Slime mould imitates development of roman roads in the balkans, *Journal of Archaeological Science Reports* 2 (2015) 264–281.
- [28] X. Zhang, A. Adamatzky, F. T. Chan, Y. Deng, H. Yang, X.-S. Yang, M.-A. I. Tsompanas, G. C. Sirakoulis, S. Mahadevan, A biologically inspired network design model, *Scientific reports* 5 (2015).
- [29] A. Adamatzky, Slime mold solves maze in one pass, assisted by gradient of chemo-attractants., *IEEE Transactions on Nanobioscience* 11 (2012) 131–4.
- [30] H. Wang, X. Lu, X. Zhang, Q. Wang, Y. Deng, A bio-inspired method for the constrained shortest path problem, *The Scientific World Journal* 2014 (2014).
- [31] X. Zhang, Y. Zhang, Y. Deng, An improved bio-inspired algorithm for the directed shortest path problem, *Bioinspiration & Biomimetics* 9 (2014).

- [32] Q. Wang, X. Lu, X. Zhang, Y. Deng, C. Xiao, An anticipation mechanism for the shortest path problem based on Physarum polycephalum, *International Journal of General Systems* 44 (2015) 326–340.
- [33] W. Jiang, Y. Luo, X. Qin, J. Zhan, An improved method to rank generalized fuzzy numbers with different left heights and right heights, *Journal of Intelligent & Fuzzy Systems* 28 (2015) 2343–2355.
- [34] Y. Zhang, Z. Zhang, Y. Deng, S. Mahadevan, A biologically inspired solution for fuzzy shortest path problems, *Applied Soft Computing* 13 (2013) 2356–2363.
- [35] X. Zhang, Q. Wang, A. Adamatzky, F. T. Chan, S. Mahadevan, Y. Deng, A biologically inspired optimization algorithm for solving fuzzy shortest path problems with mixed fuzzy arc lengths, *Journal of Optimization Theory & Applications* 163 (2014) 1049–1056.
- [36] X. Zhang, An Efficient Physarum Algorithm for Solving the Bicriteria Traffic Assignment Problem, *International Journal of Unconventional Computing* 11 (2015) 473–490.
- [37] Q. Meng, Z. Liu, Mathematical models and computational algorithms for probit-based asymmetric stochastic user equilibrium problem with elastic demand, *Transportmetrica* 8 (2012) 261–290.
- [38] E. W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik* 1 (1959) 269–271.
- [39] R. W. Floyd, Algorithm 97: Shortest path, *Communications of the Acm* 5 (1962) 345–345.
- [40] A. Adamatzky, J. Jones, Towards physarum robots: computing and manipulating on water surface, *Journal of Bionic Engineering* 5 (2008) 348–357.
- [41] Y. Liu, J. Zhang, F. Xiao, Y. Deng, A New Mutation for Traveling Salesman Problem by Physarum Polycephalum, *International Journal OF Unconventional Computing* 11 (2015) 357–373.